Supplementary Material: Distinguishing discrete and continuous behavioral variability using warped autoregressive HMMs

Julia C. Costacurta
Stanford University
jcostac@stanford.edu

Lea Duncker
Stanford University
lduncker@stanford.edu

Blue Sheffer
Stanford University

Winthrop Gillis
Harvard Medical School

Caleb Weinreb
Harvard Medical School

Jeffrey E. Markowitz
Georgia Institute of Technology, Emory University

Sandeep R. Datta
Harvard Medical School

Alex H. Williams
New York University, Flatiron Institute
alex.h.williams@nyu.edu

Scott W. Linderman
Stanford University
scott.linderman@stanford.edu

A Inference and Learning

We fit warped ARHMMs to behavioral measurements using the Expectation-Maximization (EM) algorithm. Here, we provide additional details on performing posterior inference over the latent variables $z_{1:T}$ and $\tau_{1:T}$ (E-Step), and include details on closed-form parameter updates (M-Step).

A.1 Inference

Our inference approach is the same for both model classes of WARHMM presented in the main text. Inference is performed using forward-backward message passing, with a slight twist to speed inference over a large number of (z, τ) hidden state pairs. In particular, during message passing the K discrete states and J warping variables are represented as $K \cdot J$ "paired" states, so inference via message passing becomes very slow due to multiplication with a $KJ \times KJ$ transition matrix. To ease this bottleneck we enforced Kronecker structure on the $KJ \times KJ$ transition matrix:

$$P_{(z,\tau)} = P_z \otimes P_\tau$$

Then, if we let $\alpha_t, \beta_t \in \mathbb{R}^{K \times J}$ be the forward and backward messages defined on the grid of (z, τ) values, the recursive calculations become:

$$\alpha_t = P_{(z,\tau)}\text{vec}(\alpha_{t-1} \odot l_{t-1}) \rightarrow \alpha_t = \text{vec}\left(P_z^\top(\alpha_{t-1} \odot l_{t-1})P_\tau\right)$$
$$\beta_t = P_{(z,\tau)}\text{vec}(\beta_{t+1} \odot l_{t+1}) \rightarrow \beta_t = \text{vec}\left(P_z(\beta_{t+1} \odot l_{t+1})P_\tau^\top\right),$$

where $\odot$ denotes elementwise multiplication, $l_t \in \mathbb{R}^{K \times J}$ denotes the likelihoods over the grid of (z, τ) values at time t, and $\text{vec}(\cdot)$ denotes the vectorization operation which concatenates columns of a matrix into a single column vector. This results in a complexity decrease of $O(K^2J^2)$ to $O(K^2 + J^2)$ in the matrix-vector multiplication. The posterior marginal distributions are proportional to the elementwise product, $q(z_t, \tau_t) \propto \alpha_t \odot l_t \odot \beta_t$.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

A.2 Learning

In this section, we provide additional details on the parameter updates under each model class.

A.2.1 T-WARHMM

The closed form updates for the model parameters under the T-WARHMM model class are given by

$$\mathbf{A}_k^\star = \left(\sum_{t=2}^{T} \Delta\mathbf{x}_t \mathbf{x}_{t-1}^\top\right)\left(\sum_{t=2}^{T} \mathbb{E}_{q(\tau_t|z_t=k)}\left[\frac{1}{\tau_t}\right]\mathbf{x}_{t-1}\mathbf{x}_{t-1}^\top\right)^{-1} \tag{1}$$

and

$$\mathbf{Q}_k^\star = \frac{1}{T-1}\Bigg(\sum_{t=2}^{T} \mathbb{E}_{q(\tau_t|z_t=k)}\left[\tau_t\right]\Delta\mathbf{x}_t\Delta\mathbf{x}_t^\top - \left(\mathbf{A}_k^\star\mathbf{x}_{t-1}\Delta\mathbf{x}_t^\top + \Delta\mathbf{x}_t\mathbf{x}_{t-1}^\top\mathbf{A}_k^{\star\top}\right) \tag{2}$$

$$+\mathbb{E}_{q(\tau_t|z_t=k)}\left[\frac{1}{\tau_t}\right]\mathbf{A}_k^\star\mathbf{x}_{t-1}\mathbf{x}_{t-1}^\top\mathbf{A}_k^{\star\top}\Bigg) \tag{3}$$

where $\Delta\mathbf{x}_t = \mathbf{x}_t - \mathbf{x}_{t-1}$.

A.2.2 Gaussian Process WARHMM

We place independent Gaussian priors over the transition dynamics

$$\mathbf{a}_{ijk} \sim \mathcal{N}(0, \mathbf{K}_\theta) \quad i,j = 1,\ldots,D;\ \mathbf{a}_{ijk} \in \mathbb{R}^J \tag{4}$$

The prior covariance is constructed via an exponetiated quadratic covariance function such that

$$\mathbf{K}_\theta[\tau,\tau'] = \rho^2 \exp\left(-\frac{1}{2\sigma^2}(\tau-\tau')^2\right)$$

is evaluated on the grid of τ values and $\theta = \{\rho,\sigma\}$. The full set of autoregressive dynamics for the different (z_t,τ_t) pairs, $\mathbf{A}$, is a $D\times D\times K\times J$ tensor. We will work with different reshapings of this tensor to obtain to derive a closed form update.

Let $\bar{\mathbf{A}}_\mathbf{k} \in \mathbb{R}^{D^2\times J}$ denote the reshaping of the transition dynamics. Here, each column corresponds to the vectorization of a $D\times D$ slice of the tensor, and each column corresponds to a different value on the τ grid. The prior over $\bar{\mathbf{A}}_\mathbf{k}$ can be expressed as a Matrix Normal distribution with identity row covariance I and column covariance $\mathbf{K}_\theta$:

$$\log p(\bar{\mathbf{A}}_\mathbf{k}|\theta) = -\frac{1}{2}\mathrm{Tr}[\mathbf{K}_\theta^{-1}\bar{\mathbf{A}}_\mathbf{k}^\top\bar{\mathbf{A}}_\mathbf{k}] - \frac{D^2}{2}\log|\mathbf{K}_\theta| + \text{constant} \tag{5}$$

To find the update for $\bar{\mathbf{A}}_\mathbf{k}$, we need to maximize

$$\langle P(\mathbf{x}_{1:T},\bar{\mathbf{A}}_{1:K}|\theta)\rangle_{q(\tau,z)} = -\frac{1}{2}\sum_{t=1}^{T}\left\langle(\Delta\mathbf{x}_t - \mathbf{A}_{z_t}(\tau_t)\mathbf{x}_{t-1}))^\top\mathbf{Q}_k^{-1}(\Delta\mathbf{x}_t - \mathbf{A}_{z_t}(\tau_t)\mathbf{x}_{t-1}))\right\rangle_q \tag{6}$$

$$-\frac{1}{2}\sum_{k=1}^{K}\mathrm{Tr}[\mathbf{K}_\theta^{-1}\bar{\mathbf{A}}_\mathbf{k}^\top\bar{\mathbf{A}}_\mathbf{k}] - \frac{D^2}{2}\log|\mathbf{K}_\theta| + \text{constant} \tag{7}$$

We can rewrite $\mathbf{A}_{z_t}(\tau_t)\mathbf{x}_{t-1}$ as

$$\mathrm{vec}(\mathbf{A}_{z_t=k}(\tau_t=j)\mathbf{x}_{t-1}) = (\mathbf{x}_{t-1}^\top\otimes I)(\mathbf{A}_{z_t=k}(\tau_t=j)) = (\mathbf{x}_{t-1}^\top\otimes I)\bar{\mathbf{A}}_\mathbf{k}\mathbf{1}_{\{\tau_t=j\}}$$

where $\mathbf{1}_{\{\tau_t=j\}}$ is a length-J vector with binary entries, which selects the column from $\bar{\mathbf{A}}_\mathbf{k}$ for which $\tau_t = j$.

Letting $q(z_t=k,\tau_t=j) = \omega_{kjt}$, differentiating with respect to $\bar{\mathbf{A}}_\mathbf{k}$ and setting to zero, we obtain:

$$\bar{\mathbf{A}}_\mathbf{k}\mathbf{K}_\theta^{-1} + \sum_{t=1}^{T}(\mathbf{x}_{t-1}^\top\otimes I)^\top\mathbf{Q}_k^{-1}(\mathbf{x}_{t-1}^\top\otimes I)\bar{\mathbf{A}}_\mathbf{k}\mathbf{\Omega}_{kt} = \sum_{t=1}^{T}(\mathbf{x}_{t-1}^\top\otimes I)^\top\mathbf{Q}_k^{-1}\Delta\mathbf{x}_t\omega_{kt}^\top \tag{8}$$

where $\mathbf{\Omega}_{kt} = \mathrm{diag}(\omega_{k1t}, \ldots, \omega_{kJt}) = \mathrm{diag}(\omega_{kt})$. We can show that

$$(\mathbf{x}_{t-1}^\top \otimes I)^\top \mathbf{Q}_k^{-1} (\mathbf{x}_{t-1}^\top \otimes I) = (\mathbf{x}_{t-1}\mathbf{x}_{t-1}^\top \otimes \mathbf{Q}_k^{-1}) \tag{9}$$

and

$$\sum_{t=1}^{T} (\mathbf{x}_{t-1}^\top \otimes I)^\top \mathbf{Q}_k^{-1} \Delta\mathbf{x}_t \omega_{kt}^\top = \sum_{t=1}^{T} \mathrm{vec}(\mathbf{Q}_k^{-1} \Delta\mathbf{x}_t \mathbf{x}_{t-1}^\top)\omega_{kt}^\top \tag{10}$$

Substituting these expressions into Equation (8), we obtain

$$\bar{\mathbf{A}}_\mathbf{k}\mathbf{K}_\theta^{-1} + \sum_{t=1}^{T} (\mathbf{x}_{t-1}\mathbf{x}_{t-1}^\top \otimes \mathbf{Q}_k^{-1})\bar{\mathbf{A}}_\mathbf{k}\mathbf{\Omega}_{kt} = \sum_{t=1}^{T} \mathrm{vec}(\mathbf{Q}_k^{-1}\Delta\mathbf{x}_t\mathbf{x}_{t-1}^\top)\omega_{kt}^\top \tag{11}$$

To solve this equation for $\bar{\mathbf{A}}_\mathbf{k}$ in closed form, we can rewrite the above in terms of the the length D^2J vector $\bar{\mathbf{a}}_k = \mathrm{vec}(\bar{\mathbf{A}}_\mathbf{k})$. We obtain the update equation

$$\bar{\mathbf{a}}_\mathbf{k}^\star = \left(\mathbf{K}_\theta^{-1} \otimes I_{D^2} + \sum_{t=1}^{T} \left(\mathbf{\Omega}_{kt} \otimes \mathbf{x}_{t-1}\mathbf{x}_{t-1}^\top \otimes \mathbf{Q}_k^{-1}\right)\right)^{-1} \mathrm{vec}\left(\sum_{t=1}^{T} \mathrm{vec}(\mathbf{Q}_k^{-1}\Delta\mathbf{x}_t\mathbf{x}_{t-1}^\top)\omega_{kt}^\top\right) \tag{12}$$

The update above can be expressed more efficiently in terms of expected sufficient statistics. Letting $\Sigma_t^{(0)} = \mathbf{x}_t\mathbf{x}_t^\top$, $\Sigma_t^{(1)} = \mathbf{\Delta}\mathbf{x}_t\mathbf{x}_t^\top$ and $\Sigma_t^{(2)} = \mathbf{\Delta}\mathbf{x}_t\mathbf{\Delta}\mathbf{x}_t^\top$, we obtain the expected sufficient statistics

$$\mathbf{S}^{(0)} = \Sigma_{dd't}^{(0)}\omega_{kjt} \qquad \mathbf{S}^{(1)} = \Sigma_{dd't}^{(1)}\omega_{kjt} \qquad \mathbf{S}^{(2)} = \Sigma_{dd't}^{(2)}\omega_{kjt} \tag{13}$$

where we have used tensor index notation to indicate a summation over the time index. $\mathbf{S}^{(0),(1),(2)}$ are $D \times D \times K \times J$ tensors. Precomputing and storing these statistics after each E-step allows for efficient M-Step updates.

The update for $\mathbf{Q}_k$ takes the closed form

$$\mathbf{Q}_k^\star = \frac{\sum_{j=1}^{J}\left(\mathbf{S}_{:,:,k,j}^{(2)} - \mathbf{A}_{:,:,k,j}\mathbf{S}_{:,:,k,j}^{(1)\top} - \mathbf{S}_{:,:,k,j}^{(1)}\mathbf{A}_{:,:,k,j}^\top + \mathbf{A}_{:,:,k,j}\mathbf{S}_{:,:,k,j}^{(0)}\mathbf{A}_{:,:,k,j}^\top\right)}{\sum_{j=1}^{J}\sum_{t=2}^{T}\omega_{kjt}} \tag{14}$$

B Computing details

B.1 Code, data, and instructions

The code required to reproduce our main results is available at `https://github.com/lindermanlab/warhmm`. In particular, twarhmm.py and warhmm_gp.py contain classes for the T-WARHMM and GP-WARHMM models. Each model also has a training file: train.py and train_gp.py, respectively. The main hyperparameters of each model are set in the hyperparameter_defaults variable and can be easily changed for hyperparameter sweeps using Weights and Biases [1].

The MoSeq dataset is available in combination with the original MoSeq code at the following website: `https://dattalab.github.io/moseq2-website/`. Synthetic data can be generated from the T-WARHMM using the sample() function in twarhmm.py.

B.2 Training details

All of the models shown in the paper results were trained with 50 epochs of either EM (simulated data) or stochastic EM (MoSeq dataset). The data was split 80/20 into train and test datasets.

Hyperparameters. We set specific hyperparameters as follows:

1. κ and α are as in [2] and enforce a prior on discrete state transitions, so that discrete states are more likely to remain the same for multiple time steps before switching. In our code, based on a previous sweep over α and κ we found that $\alpha = 5$ and $\kappa = 10000$ produced discrete states with durations long enough to be interpretable as behavioral syllables.

2. τ_{stay} is the probability that $\tau_t = \tau_{t+1}$, or the diagonal entry of the τ transition matrix. To approximate τ as continuous, we enforced a banded structure on the transition matrix so τ could only transition to adjacent values between time steps, and the off-diagonal values were $(1 - \tau_{\text{stay}})/2$. In our training we tried $\tau_{\text{stay}} = 0.7$ and $\tau_{\text{stay}} = 0.9$. We found that $\tau_{\text{stay}} = 0.7$ allowed for more variation of τ within discrete states, matching our desire that the warping variable could change its effect while a discrete state is carried out. Thus, $\tau_{\text{stay}} = 0.7$ was used for the paper results.
3. σ and ρ are the two parameters of the GP-WARHMM's RBF kernel. We learn values for both parameters by including them as free parameters in the EM algorithm and performing and additional "hyper-M-step" to maximize the variational lower bound.
4. C is the log step-size parameter for the T-WARHMM warping variables. In the paper experiments we took $C = 2$ based on prior assumptions about the range of speed variability in natural behavior, so that syllable instances could take on speeds from "half as fast" to "twice as fast" as the "base" syllable. Since C determines the spacing of the discretization over τ, choosing a large value for C may result in poorer algorithm performance due to large step-sizes in the state transitions from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$, while choosing a very small value would decrease the amount of variability covered in a single syllable.

Compute power. The models in section 5 (MoSeq dataset) were trained on a CPU cluster. As a comparison, a T-WARHMM with 30 discrete states and 31 warping variables took approximately 2.5 hr to train. A GP-WARHMM with the same parameters took approximately 15 hr to train. An ARHMM with 30 discrete states took approximately 1.5 hr to train.

C Societal impacts

Research on obtaining quantitative characterizations of natural behavior will have broad implications for basic research in the field of neuroscience and neuroethology. Beyond basic neuroscience research, such methods can also be used to better characterize disease phenotypes and therefore also represent an important research direction for clinical sciences. One potential negative societal impact could come if these behavioral analysis approaches are applied to human surveillance, especially as they relate to identifying the influence of drugs on behavior.

D Example video results

We include sample videos for two syllables (dart and rear) across three time warping variables (labeled slow, medium, and fast) in the supplemental material folder.

References

[1] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

[2] Alexander B Wiltschko, Matthew J Johnson, Giuliano Iurilli, Ralph E Peterson, Jesse M Katon, Stan L Pashkovski, Victoria E Abraira, Ryan P Adams, and Sandeep Robert Datta. Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 2015.